Compact preference representation and combinatorial vote

Jérôme Lang

Compact preference representation and combinatorial vote

Jérôme Lang*

Abstract

In many real-world social choice problems, the set of alternatives is defined as the Cartesian product of (finite) domain values for each of a given set of variables, and these variables cannot be asusmed to be prefentially independent (to take an example, if X is the main dish of a dinner and Y the wine, preferences over Y depends on the value taken for X). Such combinatorial domains are much too large to allow for representing preference relations or utility functions explicitly (that is, by listing alternatives together with their rank or utility); for this reason, artificial intelligence researchers have been developing languages for specifying preference relations or utility functions as compactly as possible. This paper first gives a brief survey of compact representation languages, and then discusses its role for representing and solving social choice problems, especially from the point of view of computational complexity.

1 Introduction

Voting procedures have been extensively studied by researchers in social choice theory who have studied extensively all properties of various families of voting rules, up to an important detail: candidates are supposed to be listed explicitly (typically, they are individual or lists of individuals, as in political elections), which assumes that they should not be too numerous. In this paper, we focus on the case where the set of candidates has a *combinatorial structure*, i.e., is a Cartesian product of finite value domains for each one of a set of variables: this problem will be referred to as *combinatorial vote*. In this case, the space of possible alternatives has a size being exponential in the number of variables

*IRIT - Université Paul Sabatier, Toulouse (France). lang@irit.fr

and it is therefore not reasonable asking the voters to rank or evaluate on a utility scale all alternatives.

Consider for example that the voters have to agree on a common menu to be composed if a first course dish, a main course dish, a dessert and a wine, with a choice of 6 items for each. This makes 6^4 candidates. This would not be a problem if the four items to choose were independent from the other ones: in this case, this vote problem over a set of 6^4 candidates would come down to four independent problems over sets of 6 candidates each, and any standard voting rule could be applied without difficulty. Things become more complicated if voters express dependencies between items, such as "I would like to have risotto ai funghi as first course, except if the main course is a vegetable curry, in which case I would prefer smoked salmon as first course", "I prefer white wine if one of the courses is fish and none is meat, red wine if one of the courses is meat and none is fish, and in the remaining cases I would like equally red or white wine", etc.

As soon as variables are not preferentially independent, it is generally a bad idea to decompose a combinatorial vote problem with p variables into a set of p smaller problems, each one bearing on a single variable: "multiple election paradoxes" [9] show that such a decomposition leads to suboptimal choices, and give real-life examples of such paradoxes, including simultaneous referenda on related issues. They argue that the only way of avoiding the paradox would consist in "voting for combinations [of values]", but they stress its practical difficulty: "To be sure, if there are more than eight or so combinations to rank, the voter's task could become burdensome. How to package combinations (e.g., of different propositions on a referendum, different amendments to a bill) so as not to swamp the voter with inordinately many choices – some perhaps inconsistent – is a practical problem that will not be easy to solve."

In this paper we address this issue. Since the preference structure of each voter cannot be reasonably expressed explicitly by listing all candidates, what is needed is a compact *preference representation language*. Such preference representation languages have been developed within the KR community; they are often build up on propositional logic, but not always (see for instance utility networks [1] [14] or valued constraint satisfaction [18] – however in this paper we restrict the study to logical approaches); they enable a much more concise representation of the preference structure, while preserving a good readability (and hence a proximity with the way agents express their preferences in natural language). *Therefore, the first parameter to be fixed, for a combinatorial vote problem, is the language for representing the preferences of the voters.*

Now, two other problems arise:

1. *How are these compactly represented preferences pratcically specified by the voters?* Assuming that voters can easily express by themselves (without any kind of help) their preferences over combination of values using complex logical objects

is often not reasonable; even if they do, it is highly possible that the preference relation induced by the specification is incomplete or inconsistent. So as to help agents expressing their preferences, *interactive elicitation procedures* work by finding relevant questions to ask, until the agent's preference relation is consistent and complete. Preference elicitation in combinatorial domains has been investigated in several recent works [2, 3, 13] and will not be considered here.

2. *Once preference have been elicited, how is the outcome of the voting rule computed?* Obviously, the prohibitive number of candidates makes it hard, or even practically impossible, to apply voting rules in a straightforward way, sicne all but the simplest voting procedures need a number of operations at least linear (sometimes quadratic, sometimes even exponential) in the number of candidates, which is not reasonable when the set of candidates has a strong combinatorial structure. Computational complexity of some voting procedures when applied on combinatorial domains has been investigated in [16], but this does not really address the question of *how* these procedures should be applied in practice so as to get their outcome (or an approximation of it) in a reasonable amount of time.

This article addresses the latter point.

2 Logical languages for compact preference representation

In this Section we are concerned with the preferences of a *single voter* over a finite set of candidates $\mathcal{X}$. We assume that $\mathcal{X}$ has a combinatorial nature, namely, $\mathcal{X}$ is a set of possible assignments of each of a certain number of variables to a value of its (finite) domain: $\mathcal{X} = D_1 \times ... \times D_n$, where D_i is the set of possible values for variable v_i; the size of $\mathcal{X}$ is exponentially large in n. Because specifying a preference structure explicitly in such a case is unreasonable, the AI community has developed several preference representation languages that escape this combinatorial blow up. Such languages are said to be *factorized*, or *succinct*, because they enable a much more concise representation of preference structures than explicit representations. For the sake of brevity, following we focus on *logical* languages, which means that domains are assumed to be binary. This does not imply a real loss of generality, since a variable over a finite domain with k possible values can be expressed using $\lceil \log k \rceil$ binary variables.

A *preference relation* $\succeq$ is a preorder, i.e., a reflexive and transitive binary relation on $\mathcal{A}$. $M \succeq M'$ means that alternative M is at least as good (to the agent) as alternative M'. Such a relation $\succeq$ is not necessarily complete, that is, it may be that neither $M \succeq M'$ nor $M' \succeq M$ holds for a pair of alternatives M and M' in $\mathcal{A}$. We note $M \succ M'$ for $M \succeq M'$

and not ($M \succeq M'$) (strict preference of M over M'), and $M \sim M'$ for $M \succeq M'$ and $M' \succeq M$ (indifference). It is important to note that $M \sim M'$ means that the agent takes M and M' to be equally preferred, while the incomparability between M and M' ($M \not\succeq M'$ and $M' \not\succeq M$) simply means that no preference between them is expressed.

These definitions are about preferences over an arbitrary set of alternatives $\mathcal{A}$. In this paper, we consider propositional languages expressing preferences: such languages express preferences over the set of possible interpretations W over a given alphabet VAR. A refinement of this definition is that of assuming that the set of possible alternatives excludes some interpretations of W. In this case, we assume that a formula K is given: this formula represents "integrity constraints" on the set of *feasible* alternatives, i.e., the only interpretations we accept as possible alternatives are those of $Mod(K)$, i.e., $\mathcal{A} = Mod(K)$. For instance, in a decision making problem consisting of recruiting at least one and at most two of three candidates a, b and c, the feasible alternatives are the models of $K = (a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$.

We now briefly recall the propositional languages for preference representation we study. In the following, the formulas G_i are propositional formulas representing elementary *goals*. The input of a logically-represented preference relation is a pair $\Delta = \langle K, GB \rangle$ where K is the propositional formula restricting the possible alternatives (the integrity constraints) and GB (the *goal base*) is a set of elementary goals, generally associated with extra data such as weights, priorities, contexts or distances. $\succeq_{K,GB}$ (or simply $\succeq_{GB}$ when there is no risk of ambiguity) denotes the preference relation induced by GB over $Mod(K)$.

2.1 A brief overview of languages

2.1.1 Penalties

In this natural and frequently used preference representation language, the agent expresses her preferences in terms of a set of propositional formulas that she wants to be satisfied. In order to compare alternatives (models), formulas are associated with weights (usually, numbers), which tell how important the satisfaction of the formula is considered. Formally, the preferences of an agent are expressed as a finite set of goals, where each goal is a propositional formula with an associated weight. The complete preference is given by a set of these goals: $GB = \{\langle \alpha_1, G_1 \rangle, \ldots, \langle \alpha_n, G_n \rangle\}$, where each α_i is an integer and each G_i is a propositional formula. The degree of preference of a model is measured as follows: for any $M \in Mod(K)$, we define $p_{GB}(M) = \sum \{\alpha_i | M \not\models G_j\}$ to be the penalty of M. The preference relation $\succeq_{GB}^{pen}$ is defined by $M \succeq_{GB}^{pen} M'$ if and only if

$p_{GB}(M) \leq p_{GB}(M')$ (with the convention $\sum(\emptyset) = 0$).[1]

2.1.2 Distance to goals

The preference relation based on penalties only makes a distinction between models satisfying a formula and models violating it. On the other hand, if an agent prefers a formula G_i to be satisfied, we could infer that she also prefers models "close" to this formula than models "far". Let d be a pseudo-distance between models, that is, a symmetric function from $\mathcal{X}^2 \rightarrow \mathbb{R}$ such that $d(M, M') = 0$ if and only of $M = M'$. For instance, the Hamming distance $d_H(M, M')$ is the number of variables that are assigned different values in M and M'.) The "distance" between a model M and a formula G is defined by $d(M, G) = \min_{M' \models G} d(M, M')$. A goal base is a finite set of pairs $\langle \alpha_i, G_i \rangle$; the distance of a model to a goal base is defined by $d(M, GB) = \sum_i \{\alpha_i.d(M, G_i)\}$. and finally, $\succeq^H_{GB}$ is defined by

$$M \succeq^H_{GB} M' \text{ if and only if } d(M, GB) \leq d(M', GB)$$

2.1.3 Prioritized Goals

The languages defined above allow for compensations among goals (the violation of a goal may be compensated by the satisfaction of a sufficient number of goals of lower importance). Prioritization is used when such a compensation should not be possible, and does not need any numerical data. In this case, a goal base is a pair $GB = \langle \{G_1, \ldots, G_n\}, r \rangle$ where each G_i is a propositional formula and r is a rank function from $\{1, \ldots, n\}$ to $\mathbb{N}$: if $r(i) = j$, then j is called the rank of the formula G_i. By convention, a lower rank means a higher priority. The question is now how to extend the priority on goals to a preference relation on alternatives. The following three choices are the most frequent ones:

best-out ordering Let $r_{GB}(M) = \min\{r(i) \mid M \not\models G_i\}$ Then $M \succeq^{bo}_{GB} M'$ iff $r_{GB}(M) \geq r_{GB}(M')$

discrimin ordering Let $discr^+_{GB}(M, M') = \{i \mid M \models G_i \text{ and } M' \not\models G_i\}$ and $discr_{GB}(M, M') = discr^+_{GB}(M, M') \cup discr^+_{GB}(M', M)$ Then:

$$\left| \begin{array}{l} M \succ^{discrimin}_{GB} M' \text{iff} \min_{i \in discr^+_{GB}(M,M')} r(i) < \min_{j \in discr^+_{GB}(M',M)} r(j) \\ M \succeq^{discrimin}_{GB} M' \text{iff} M \succ^{discrimin}_{GB} M' \text{ or } discr_{GB}(M, M') = \emptyset. \end{array} \right.$$

[1]Many other operators can be used, in place of the sum, for aggregating weights of violated (or symmetrically, satisfied) formulas (see [15] for a general discussion).

leximin ordering Let $d_k(M)$ be the cardinal of $\{i \mid M \models G_i$ and $r(i) = k\}$.

$$\left| \begin{array}{l} M \succ_{GB}^{leximin} M' \text{iff} \exists k \leq n \text{ s. t.} d_k(M) > d_k(M') \text{ and } \forall j < k, d_k(M) = d_k(M'); \\ M \succeq_{GB}^{leximin} M' \text{iff} M \succ_{GB}^{leximin} M' \text{ or } d_i(M) = d_i(M') \text{ for any } i. \end{array} \right.$$

Note that $\succeq_{GB}^{leximin}$ and $\succeq_{GB}^{bo}$ are complete preference relations while $\succeq_{GB}^{discrimin}$ is generally not. We moreover have the following chain of implications: $M \succ_{GB}^{bo} M' \Rightarrow M \succ_{GB}^{discrimin} M' \Rightarrow M \succ_{GB}^{leximin} M'$.

More discussion, references and examples can be found in [16, 10].

2.1.4 Ceteris Paribus preferences

In this language, preferences are expressed in terms of statements like: "all other things being equal, I prefer these alternatives over these other ones." Formally, let C, G, and G' be three propositional formulas and V being a subset of VAR such that $Var(G) \cup Var(G') \subseteq V$. The *ceteris paribus desire* $C : G > G'[V]$ means: "all irrelevant things being equal, I prefer $G \wedge \neg G'$ to $\neg G \wedge G'$", where the "irrelevant things" are the variables that are not in V. The definitions proposed in various places differ somehow – we take here the definition of [10]. For natural reasons, and to remain consistent with the original definitions, we impose that $Var(G) \cup Var(G') \subseteq V$.

Furthermore, we add to the original definition the ability to express *indifference statements* – without them, $M \sim M'$ could not be expressed.

Let $GB = \mathcal{D}_P \cup \mathcal{D}_I$, where $\mathcal{D}_P$ and $\mathcal{D}_I$ are defined as follows.

$$\begin{array}{rcl} \mathcal{D}_P & = & \{C_1 : G_1 > G'_1[V_1], \ldots, C_m : G_m > G'_m[V_m]\} \\ \mathcal{D}_I & = & \{C_n : G_n \sim G'_n[V_n], \ldots, C_p : G_p \sim G'_p[V_p]\} \end{array}$$

We call the elements of $\mathcal{D}_P$ as "preference desires" while elements of $\mathcal{D}_I$ are "indifference desires". For all i, C_i, G_i and G'_i are propositional formulas and $Var(G_i) \cup Var(G'_i) \subseteq V_i \subseteq VAR$. We define the preference induced by a single desire $D_i = C_i : G_i > G'_i[V_i]$, denoted by $M >_{D_i} M'$, by the following three conditions:

1. $M \models C_i \wedge G_i \wedge \neg G'_i$;
2. $M' \models C_i \wedge \neg G_i \wedge G'_i$;
3. M and M' coincide on all variables in $VAR \backslash V_i$.

If the above conditions 1-3 are satisfied for an indifference desire $D_i = C_i : G_i \sim G'_i[V_i]$ in $\mathcal{D}_I$, then we say that M and M' are *indifferent* with respect to D_i, denoted by $M \sim_{D_i} M'$. Now, the preference order $\succeq^{cp}_{GB}$ is defined from the above dominance relations by transitive closure of their union: $M \succeq^{cp}_{GB} M'$ holds if and only if there exists a finite chain $M_0 = M, M_1, \ldots, M_{q-1}, M_q = M'$ of alternatives such that for all $j \in \{0, \ldots, q-1\}$ there is a $D_i \in GB$ such that $M_j >_{D_i} M_{j+1}$ or such that $M_j \sim_{D_i} M_{j+1}$.

An important sublanguage of CP-preferences is the language of (binary) **CP-nets**, which is obtained by imposing the following syntactical restriction:

- goals G and G' are *literals*, that is, CP-statements express preference of a value over its opposite for a given single variable, given some context (in other words, G and G' are of the form $(\mathbf{x}_i = v_i)$, where $\mathbf{x}_i \in VAR$ and $v_i \in \{T, F\}$.
- the variables mentioned in the context C of a preference statement about variable $\mathbf{x}_i$ must be contained in a fixed set of variables, called the *parents of* $\mathbf{x}_i$, denoted by $Parents(\mathbf{x}_i)$.
- for each variable $\mathbf{x}_i$ and each possible assignment π of the parents of $\mathbf{x}_i$, there is *one and only one* CP-preference $C : x_i > \neg x_i$ or $C : \neg x_i > x_i$ such that $\pi \models C$.

The more expressive language of **TCP-nets** [7] can also be obtained by syntactical restrictions. See [19] for a discussion about the expressivity of these various languages.

For the sake of brevity, we omitted the family of preference representation languages based on *conditional logics*. See [16, 10].

2.2 Issues in preference representation

At least four very important problems must be addressed when investigating the relevance and complexity of preference representation languages.

Elicitation We already discussed this issue in Introduction and we do not want to come back on this, since since is left outside the scope of this paper.

Expressive power R being a representation language, a relevant quesiton is whether R can express all preorders and/or all utility functions, or only complete preorders, or only a strict subclass of them, etc. This issue is investigated in [10].

Computational complexity Let R being representation language. What is the computational complexity of comparing two alternatives given an input GB of R, of deciding whether a given alternative is optimal, of finding an optimal alternative? This issue is investigated in [16].

Comparative succinctness Given R, R' two representation languages, R' is said to be at least as succinct as R iff is there a function F from R to R' such that

a. for each $GB \in R$, GB and $F(GB)$ induce the same preference relation (or utility function);

b. F is polysize, i.e., there exists a polynomial function p such that for all $GB \in R$, $size(F(GB)) \leq p(size(GB))$.

This issue is investigated in [10].

3 Combinatorial vote

Let $\mathcal{A} = \{1, ..., N\}$ be a finite set of *voters*; $\mathcal{X}$ is a *finite* set of *alternatives* (or candidates); a *individual preference profile* P is a complete weak order $\succeq_i$ (reflexive and transitive relation) on $\mathcal{X}$. A *preference profile* w.r.t. $\mathcal{A}$ and $\mathcal{X}$ is a collection of N individual preference profiles: P = ($\succeq_1$, ..., $\succeq_N$). Lastly, let $\mathcal{P}_{A,X}$ set of all preference profiles.

A *voting correspondance* $C : P_{A,X} \to 2^{\mathcal{X}} \setminus \{\emptyset\}$ maps each preference profile P of $P_{A,X}$ into a nonempty subset $C(P)$ of $\mathcal{X}$. A *voting (deterministic) rule* $r : P \to \mathcal{X}$ maps each preference profile P of $P_{A,X}$ into a single candidate $r(P)$. A deterministic rule can be obtained from a correspondance by prioritization over candidates (for more details see [8]). In the rest of the paper we focus on deterministic rules.

A combinatorial vote problem consists in applying voting rules when the set of alternatives has a combinatorial structure and the voters' preferences are expressed in a compact preference representation language. Practically, a combinatorial vote problem is composed of two steps: first, the agents express their preferences within a fixed (and common) representation language R, and second, one or several optimal (i.e., non-dominated) candidate(s) is (are) determined automatically, using a fixed voting rule.

For any representation language R, one defines a *R-profile for p voters* as a collection $B = \langle GB_1, ..., GB_p\rangle$ of goal bases (one for each of the p voters), expressed in the language R, generating a profile $P = Induce_R(B) = \{\succeq_{GB_1}, ..., \succeq_{GB_p}\}$.

3.1 Combinatorial vote: direct approach

The "direct" approach to solving a combinatorial vote consists in applying these tasks in sequence:

- elicit the preference relation for each voter, using a compact representation language;

- generate the whole preference relations on $D_1 \times \ldots \times D_n$ from the input;
- apply the voting rule r.

The good point with this direct approach is that it leads to finding an optimal outcome, more precisely, it allows for determining the exact winners according to the chosen voting rule and the true preference of the agents. The (very) bad point is its very high computational complexity in the general case. Here are examples, in the simplest compact representation language, that is, the basic propositional representation (where each agent specifies a unique propositional formula as his/her goal):

- computing a winner for the plurality rule needs $O(\log N)$ satisfiability tests (N = number of agents);
- determining whether there exists a Condorcet winner is both NP-hard and coNP-hard, and in in Θ_2^p (the exact complexity is an open problem).

Further results, including for instance the complexity of determining whether there exists a Condorcet winner for a given profile specified in a compact preference representation language, can be found in [16].

3.2 Combinatorial vote: sequential approach

The principle of the sequential appraoch is to exploit preferential independence of the preference profiles. It is well-known that preferences relations (or utility functions) over combinatorial "real-life" domains most often enjoy structural properties such as *(conditional) preferential independence* between sets of variables. This assumption was central to the development of several preference representation languages, especially *graphical languages* such as CP-nets of weighted constraint satisfaction. In these languages, the input consists of two distinct part: a structural part (an hypergraph in the CSP case, a directed acyclic graph in the CP-net case) over the variables, and a "internal" part consisting of the local preference relations over the subsets of variables identified by the structural part.

For instance, let $V = \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t}\}$, all three being Boolean variables, and assume that preference of a given agent over 2^V can be defined by a CP-net whose structural part is the directed acyclic graph $G = \{(x, y), (y, z), (y, t), (z, t)\}$; this means that, for the agent considered, preference over the values of **x** is unconditional, preference over the values of **y** is fully defined given the value of **x**, and so on.

Now comes the central assumption to the sequential approach to combinatorial vote: *the preferential independence structure is common to all agents*. Therefore, for instance, if preference obver 2^V for agent 1 can be described by a CP-net with the structure as above, then all other agents are assumed to be able to express their preferences within a CP-net using the same structure. This is a strong assumption; however, in many real-life domains it can be considered as reasonable.

Let us first consider an example. Let $N = 7$, $V = \{\mathbf{x}, \mathbf{y}\}$ with $Dom(\mathbf{x}) = \{x, \bar{x}\}$ and $Dom(\mathbf{y}) = \{y, \bar{y}\}$, and let us consider the following preference relations, where each agent expresses his preference relation by a CP-net corresponding to the following fixed preferential structure: preference on **x** is unconditional (but preference on **y** may depend on the value given to **x**).

3 agents	2 agents	2 agents
$\bar{x} \succ x$	$x \succ \bar{x}$	$x \succ \bar{x}$
$x : \bar{y} \succ y$	$x : y \succ \bar{y}$	$x : \bar{y} \succ y$
$\bar{x} : y \succ \bar{y}$	$\bar{x} : \bar{y} \succ y$	$\bar{x} : y \succ \bar{y}$

This corresponds to the following preference relations:

3 agents	2 agents	2 agents
$\bar{x}y$	xy	$x\bar{y}$
$\bar{x}\bar{y}$	$x\bar{y}$	xy
$x\bar{y}$	$\bar{x}\bar{y}$	$\bar{x}y$
xy	$\bar{x}y$	$\bar{x}\bar{y}$

Let r be a deterministic rule r. Since for all 7 voters, preference on **x** is unconditional, we may consider first the projections of the 7 preference relations on $Dom(\mathbf{x})$, namely $\langle P_1^{\mathbf{x}}, \ldots, P_n^{\mathbf{x}} \rangle$, and start by applying r to these, which results in a value of **x**, denoted by x^*, called the **x**-winner[2]. The value of **x** is now fixed to x^*; then, let us consider the projections of the 7 preference relations on $Dom(\mathbf{y})$, given $\mathbf{x} = x^*$; denote these by $\langle P_1^{\mathbf{y}|\mathbf{x}=x^*}, \ldots, P_n^{\mathbf{y}|\mathbf{x}=x^*} \rangle$; we then apply r to these, which results in a value of **y**, denoted by y^*, called the conditional $\mathbf{y}$-winner given $\mathbf{x} = x^*$. The *sequential winner* is now obtained by combining the **x**-winner and the conditional $\mathbf{y}$-winner given $\mathbf{x} = x^*$, namely (x^*, y^*).

Example: let r be the plurality rule (where the plurailty score of a candidate is the number of voters ranking this candidate in the highest position, the plurality winners then being those maximizing the plurality score). Because 4 agents out of 7 unconditionally prefer x over $\bar{x}$, we get $x^* = x$; then, given **x** = x, 5 agents out of 7 prefer $\bar{y}$ to y, which leads to $y^* = x$. Therefore, the sequential plurality winner is (x, y). However, the direct plurality winner is $(\bar{x}, y)$.

[2]In case of ties, we therefore need a deterministic tiebreaking mechanism, for instance using a predeterminate order over the possible values of **x**).

The above example shows that when r is the plurality rule, sequential winners (obtained by sequential applications of r) and direct winners (obtained by a direct application of r) do not always coincide, which is an argument against the use of the sequential approach for such a voting rule. Note that more generally, this failure of sequential winners to coincide with direct winners holds for any scoring rule.

A more general question is the following: are there deterministic rules r, does the sequential winner (obtained by sequential applications of r) and the direct winner (obtained by a direct application of r) coincide? We do not know any positive answer to this question in the general case. We first show a second negative result, and lastly we give a restriction on preferences under which the answer to the above question turns out to be positive.

Here comes the second negative result. A Condorcet winner is a candidate preferred to any other candidate by a majority of voters. The notion of Condorcet winner naturally leads to the determination of *sequential Condorcet winners*: let X and Y being two subsets of the set of variables; then

- if preference on X is unconditional, then $\vec{x} \in D_X$ is a X-Condorcet winner if and only if
$$(\forall \vec{y} \in D_{\bar{X}}) \forall \vec{x}' \in D_X \ \#\{i, \vec{x}\vec{y} \succ_i \vec{x}'\vec{y}\} > \frac{N}{2}$$
- if and preference on Y given X is unconditional, then $\vec{y} \in D_Y$ is a Y-Condorcet winner given $X = \vec{x}$ if and only if
$$(\forall \vec{z} \in D_{\overline{X \cup Y}}) \forall \vec{y}' \in D_X \ \#\{i, \vec{x}\vec{y}\vec{z} \succ_i \vec{x}\vec{y}'\vec{z}\} > \frac{N}{2}$$

The sequential Condorcet winner is then the sequential combination of "local" Condorcet winners. The question is now, is a sequential Condorcet winner a direct Condorcet winner and vice versa? The following example shows that this fails.

2 voters	1 voter	2 voters
$x\bar{y}$	xy	$\bar{x}y$
$\bar{x}\bar{y}$	$x\bar{y}$	$\bar{x}\bar{y}$
xy	$\bar{x}y$	xy
$\bar{x}y$	$\bar{x}\bar{y}$	$x\bar{y}$

x and **y** are preferentially independent, therefore the sequential Condorcet winner is the mere combination of the local Condorcet winner for $\{$**x**$\}$ and the local Condorcet winner for $\{$**y**$\}$, provided that both exist. Since 3 voters unconditionally prefer x to $\bar{x}$, x is the $\{\mathbf{x}\}$-Condorcet winner; similarly, 3 voters unconditionally prefer y to $\bar{y}$ and is the $\{\mathbf{y}\}$-Condorcet winner. Therefore, xy is the sequential Condorcet winner – *but* xy is not a direct Condorcet winner, because four voters out of seven prefer $\bar{x}\bar{y}$ to xy.

We now give a condition on the preference relations such that direct and sequential Condorcet winners coincide. We say that a preference relation on $Dom(\mathbf{x}_1 \times \ldots \times Dom(\mathbf{x}_p\}$ is *lexicographic* if and only if there is a total ordering of the variables, say without loss of generality $\mathbf{x}_1 \rhd \mathbf{x}_2 \rhd \ldots \rhd \mathbf{x}_p$, and p local preference relations on $Dom(\mathbf{x}_1)$, …, $Dom(\mathbf{x}_p)$, such that $x = (x_1, \ldots, x_p)$ is preferred to $y = (y_1, \ldots, y_p)$ iff there is an index $j \leq p$ such that (a) for every $k \leq j$, $x_k \sim y_k$ and (b) $x_j \succ y_j$. Now, assume that all agents have lexicographic preference relations (with the same variable ordering). $(v_1, \ldots, v_p) \in D_1 \times \ldots D_p$ is a sequential Condorcet winner iff

- $v_1 \in D_1$: $\{\mathbf{x}_1\}$-Condorcet winner;
- $v_2 \in D_2$: $\{\mathbf{x}_2\}$-Condorcet winner given $\mathbf{x}_1 = v_1$;
- …
- $v_p \in D_p$: $\{\mathbf{x}_p\}$-CW given $\mathbf{x}_1 = v_1, \ldots, \mathbf{x}_{p-1} = v_{p-1}$

Thenwe have the following positive result: if there exists a sequential Condorcet winner $(v_1, \ldots, v_p)$ then $(v_1, \ldots, v_p)$ is also the (direct) Condorcet winner for the given profile, and *vice versa*.

Now, the restriction on lexicographic preference relation is a strong one. Thie leads to the following questions and problems:

Question 1 are there voting rule such that sequential winners and direct winners always coincide?

Problem 2 find reasonable restrictions on the preference relations so that the answer to Question 1 becomes positive;

Problem 3 find good algorithms (using the preferential structure) for determining winners of a combinatorial vote problem.

References

[1] Fahiem Bacchus and Adam Grove. Graphical models for preference and utility. *Proceedings of the 11th Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, 3-10.

[2] Craig Boutilier. A POMDP formulation of preference elicitation problems. P*Proceedings of AAAI-02*, AAAI Press, 2002.

[3] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos and David Poole. Preference-Based Constrained Optimization with CP-Nets. *Computational Intelligence* 20 (2), 137-157, 2004.

[4] Craig Boutilier, Ronen Brafman, Holger Hoos and David Poole. Reasoning with conditional *ceteris paribus* statements. *Proceedings of the 15th Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, 71-80.

[5] Craig Boutilier, Fahiem Bacchus and Ronen Brafman. UCP-networks: a directed graphical representation of conditional utilities. *Proceedings of UAI-2001*, 56-64.

[6] Craig Boutilier and Holger Hoos. Bidding languages for combinatorial auctions. *Proceedings if IJCAI'01*, 1211-1217.

[7] Ronen I. Brafman, Carmel Domshlak and Eyal Shimony. Introducing variable importance tradeoffs into CP-nets. *Proceedings of the 18th Intern. Conf. on Uncertainty in Artificial Intelligence (UAI-2002)*, Morgan Kaufmann, 2002.

[8] Steven J. Brams and Peter C. Fishburn. Voting procedures. *Handbook of Social choice and Welfare*, chapter 4, Elsevier, 2004.

[9] Steven J. Brams, D. Marc Kilgour and William S. Zwicker. The paradox of multiple elections. *Social Choice and Welfare* 15: 211-236, 1998.

[10] Sylvie Coste-Marquis, Jérôme Lang, Paolo Liberatore and Pierre Marquis. Expressivity and succinctnedd of preference representation languages. *Proceedings of the 9th Internationl Conference on Knowledge Representation and Reasoning (KR-2004)*, 203-212, AAAI Press, 2004.

[11] Vincent Conitzer and Tuomas Sandholm. *Vote elicitation: complexity and strategy-proofness*. *Proceedings of AAAI-02*.

[12] Carmel Domshlak, Ronen Brafman and S. Eyal Shimony. Preference-Based Configuration of Web Page Content. *Proc. of IJCAI'2001*.

[13] Christophe Gonzales and Patrice Perny. Graphical models for utility elicitation. *In these proceedings* (preliminary version in *Proceedings of the 9th Internationl Conference on Knowledge Representation and Reasoning (KR-2004)*, 224-233, AAAI Press, 2004.

[14] Pierfrancesco La Mura and Yoav Shoham. Expected utility networks. *Proceedings of UAI'99*, 366-373, 1999.

[15] C. Lafage, J. Lang. Logical representation of preferences for group decision making. *Proceedings of KR'2000*, 457-468, 2000.

[16] Jérôme Lang. Logical Preference Representation and Combinatorial Vote *Annals of Mathematics and Artificial Intelligence* vol. 42, no. 1-3, pp. 37-71(35), 2004.

[17] Noam Nisan. Bidding and allocations in combinatorial auctions. In *ACM Conference on Electronic Commerce (EC-2000)*, Minneapolis, 2000.

[18] Thomas Schiex, Hélène Fargier and Gérard Verfaillie. Valued constraint satisfaction problems: hard and easy problems, *Proceedings of IJCAI'95*, 631-637, 1995.

[19] Nic Wilson. Extending CP-nets with stronger conditional preference statements. *Proceedings of AAAI-04*, 2004.